

CLAIMS

1. A method of processing requests for performing operations on files which are capable of being stored in a UNIX file system, said method comprising:

receiving a request to perform at least one operation on a file stored in said

UNIX file system;

determining whether at least one mandatory lock is associated with said file;

determining a mandatory lock category for said at least one mandatory lock when said determining determines that at least one mandatory lock is associated with said file;

determining whether said at least one operation should be allowed for at least one mandatory lock; and

allowing said at least one operation when said determining determines that said at least one operation should be allowed.

2. A method a recited in claim 1, wherein said method further comprises:

denying said at least one operation when said determining determines that said at least one operation should not be allowed.

3. A method a recited in claim 1, wherein said request is sent by a Windows client.

4. A method a recited in claim 3, wherein said Windows client is a Common Internet File System client.

5. A method a recited in claim 1, wherein said method further comprises:

determining at least one mandatory lock type for said at least one mandatory lock when said determining determines that at least one mandatory lock is associated with said file.

6. A method a recited in claim 1,

wherein said method further comprises:

determining at least one mandatory lock type for said at least one mandatory lock when said determining determines that at least one mandatory lock is associated with said file;

wherein said determining of whether said at least one operation should be allowed for said at least one mandatory lock comprises:

determining whether said least one mandatory lock type is compatible with said at least one operation.

5

7. A method as recited in claim 1, wherein said at least mandatory lock category can be a Byte-Range lock or a Shared Resource lock.

10

8. A method as recited in claim 7, wherein the type of said Byte-Range lock can be exclusive or shared.

9. A method as recited in claim 7,
wherein said Shared Resource lock can have a deny mode associated with it;
and
wherein said deny mode can be defined with respect to reading or writing of said file.

10. A method as recited in claim 1, wherein said at least one operation can be a read, write, delete, rename, memory map or change size operation.

11. A method of processing requests to perform operations on a file which is capable of being stored in a UNIX file system, said method comprising:

(a) receiving a request to perform an operation on a UNIX file which has a mandatory Byte-Range lock associated with it;

25 (b) determining whether said requested operation may affect a byte range of the UNIX file; said byte range representing a portion of said file which is associated with the mandatory Byte-Range lock; and

30 (c) determining whether said operation is compatible with said Byte-Range lock when said determining (b) determines that said requested operation may affect said byte range.

12. A method as recited in claim 11, wherein said method further comprises:
determining whether said request was made by the owner of said Byte-Range lock when said determining (c) determines that said operation is not compatible with said Byte-Range.

5

13. A method as recited in claim 11, wherein said mandatory Byte-Range lock can be an exclusive or shared lock.

10

14. A method as recited in claim 11, wherein said requested operation is initiated from a Windows environment.

15. A method of processing requests to perform operations on a file which is capable of being stored in a UNIX file system, said method comprising:

(a) receiving a request to perform an operation on a UNIX file which has a mandatory Shared Resource lock associated with it; and

(b) determining whether a deny mode associated with the file covers an access mode associated with said request for accessing said file.

16. A method a recited in claim 15, wherein said method further comprises:

determining whether said request was made by the owner of said mandatory Shared Resource lock when said determining (b) determines that said deny mode associated with the file covers said access.

25

17. A method as recited in claim 15, wherein said deny and access modes can be defined with respect to at least one the operations of reading from and writing to files.

18. A method as recited in claim 15, wherein said requested operation is initiated from a Windows environment.

30

19. A method of processing a request to change the size of a file which is capable of being stored in a UNIX file system; said method comprising:

determining whether a mandatory Byte-Range lock or a mandatory Shared Resource lock is associated with said file;

determining whether said Shared Resource lock includes a deny write operation when said determining determines that a mandatory Shared Resource lock is associated with said file; and

identifying a region of said file which may be affected by said request to
5 change the size of said file when said determining determines that a mandatory Byte-Range lock has been associated with the file.

20. A method as recited in claim 19, wherein said method further comprises:

determining whether said identified region intersects a locked region of said
10 file; and

allowing said request to change said file size when said determining determines that said identified region does not intersect said locked region of said file.

21. A method as recited in claim 19, wherein said method further comprises:

determining whether said request was made by the owner of said mandatory Byte-Range lock or mandatory Shared Resource lock.

22. A method as recited in claim 19, wherein said request to change said file size is allowed when said determining whether said Shared Resource lock does not include a deny write operation.

23. A UNIX computing environment, comprising:

a UNIX file system capable of storing one or more files therein;

25 wherein said UNIX computing environment is capable of enforcing mandatory locks for said one or more files in said UNIX file system; and

wherein said mandatory locks comprise of at least two mandatory lock categories.

30 24. A UNIX computing environment as recited in claim 23, wherein said computing environment further comprises:

a distributed file system;

a file system-independent portion of an operating system; and

wherein said mandatory locks are at least partially implemented in said distributed file system and said file system-independent portion of said operating system.

5 25. A UNIX computing environment as recited in claim 24, wherein said at least two categories comprise Byte-Range locks and Shared Resource locks.

10 26. A UNIX computing environment as recited in claim 23, wherein said mandatory locks can be enforced with respect to read, write, delete, rename, memory map, or change size operations.

27. A computer readable media including computer program code for performing operations on files which are capable of being stored in a UNIX file system, said computer readable media comprising:

computer program code for receiving a request to perform at least one operation on a file stored in said UNIX file system;

computer program code for determining whether at least one mandatory lock is associated with said file;

computer program code for determining a mandatory lock category for said at least one mandatory lock when said determining determines that at least one mandatory lock is associated with said file;

computer program code for determining whether said at least one operation should be allowed for at least one mandatory lock; and

25 computer program code for allowing said at least one operation when said determining determines that said at least one operation should be allowed.

28. A computer readable media as recited in claim 27, wherein said computer readable media further comprises:

30 computer program code for denying said at least one operation when said determining determines that said at least one operation should not be allowed.

29. A computer readable media as recited in claim 27, wherein said request is sent by a Windows client.